

AMENDMENTS TO THE CLAIMS

Please amend the claims as follows:

1. (Currently Amended) A computer-readable storage medium having stored thereon a data structure, the data structure separating storage of an attribute value from handling of the attribute value, comprising:

a) a model element class configured to implement the constructs described by metadata; the model element class storing an attribute value in a private member field of the model element class in the same memory block as a declaring class;

b) a nested handler class, wherein the nested handler class is a subclass of a generic handler class and inherits base functionality from the generic handler class, wherein the nested handler class has public access to an enclosing element's private members stored in the private member field of the model element class;

c) a meta-attribute information object configured to describe attributes of the model element class;

d) a singleton model element field handler object comprising a singleton pattern as an abstract base class for handling inlined field values of the model element class configured to access the attribute value stored in the model element class, wherein a first subclass of the singleton model element field handler object comprises a typed model element field handler subclass defining a get value function;

e) a base filed handler class which acts as an intermediary generic mechanism when getting the filed values in the model element filed handler object, wherein the get value function is configured to:

i) dispatch to a sub-classed get field value function in the first subclass which is type safe;

ii) access the model element class and return the attribute value directly upon request; and

iii) provide an entry point for non-type safe application programming interface so that general purpose client code can access the filed values without relying on type safety; and

wherein the storage of the attribute value is separate from handling of the attribute value.

2. (Canceled)

3. (Canceled)

4. (Original) The computer-readable medium of claim 1, wherein the model element field handler object sets the attribute value sorted in the model element class.

5. (Canceled)

6. (Canceled)

7. (Previously Presented) The computer-readable medium of claim 1, wherein the typed model element field handler subclass defines a set value function for setting the attribute value.

8. (Original) The computer-readable medium of claim 1, wherein the data structure further comprises

d) a meta-class information object for storing data associated with the model element.

9. (Currently Amended) A computer-readable storage medium having having stored thereon a data structure, the data structure separating storage of an attribute value from handling of the attribute value, comprising:

- a) a container for storing meta-data in a tree structure;
- b) a model element class configured to implement the constructs described by metadata; the model element class storing an attribute value in a private member field of the model element class in the same memory block as a declaring class;
- c) a nested handler class, wherein the nested handler class is a subclass of a generic handler class and inherits base functionality from the generic handler class, wherein the nested handler class has public access to an enclosing element's private members stored in the private member field of the model element class;
- d) a meta-class information object configured to store data associated with the model element;
- e) a meta-attribute information object configured to describe attributes of the model element class;
- f) a singleton model element field handler object comprising a singleton pattern as an abstract base class for handling inlined field values of the model element class configured to access the attribute value stored in the model element class, wherein a first subclass of the model element field handler object comprises a typed model element field handler subclass defining a set value function for setting the attribute value;
- g) a base filed handler class which acts as an intermediary generic mechanism when setting the filed values in the model element filed handler object, wherein the set value function is configured to:

- i) provide validation of a new value;
- ii) record necessary undo information; and

iii) dispatch to a sub-classed set field value function in the first subclass which is type safe;

wherein the nested handler class is configured to directly access data in the model element class as the nested handler class has public access to the private members of the model element class; and

wherein the storage of the attribute value is separate from handling of the attribute value.

10. (Previously Presented) The computer-readable medium of claim 9, wherein the container comprises a store acting as a root of the tree structure.

11. (Canceled)

12. (Original) The computer-readable medium of claim 9, wherein the model element field handler object sets the attribute value stored in the model element class.

13. (Canceled)

14. (Previously Presented) The computer-readable medium of claim 9, wherein the typed model element field handler subclass defines a get value function for accessing the attribute value.

15. (Canceled)

16. (Currently Amended) A method implemented at least in part by a computing device, the computing device accessing an attribute value within a data structure, the data

structure separating storage of the attribute value from handling of the attribute value, the method comprising:

a) storing the attribute value in a private member field of a model element class in a same memory block as a declaring class;

b) declaring a nested handler class, the nested handler class being a subclass of a generic handler class and inherits base functionality from the generic handler class, wherein the nested handler class has public access to an enclosing element's private members stored in the private member field of the model element class;

c) providing a singleton model element field handler object comprising a singleton pattern as an abstract base class for handling inlined field values of the model element class configured to access the attribute value stored in the model element class, wherein a first subclass of the singleton model element field handler object comprises a typed model element field handler subclass defining a get value function;

d) issuing the get value function to obtain the attribute value from the model element class; and

e) receiving the attribute value from the model element class;

f) providing a base filed handler class which acts as an intermediary generic mechanism when getting the filed values in the model element filed handler object, wherein the get value function is configured to:

i) dispatch to a sub-classed get field value function in the first subclass which is type safe;

ii) access the model element class and return the attribute value directly upon request; and

iii) provide an entry point for non-type safe application programming interface so that general purpose client code can access the filed values without relying on type safety;

~~wherein the get value function is configured to access the model element class and to return the attribute value directly upon request; and~~

wherein the storage of the attribute value is separate from handling of the attribute value.

17. (Canceled)

18. (Currently Amended) A method implemented at least in part by a computing device, the computing device setting an attribute value within a data structure, the data structure separating storage of the attribute value from handling of the attribute value, the method comprising:

a) declaring a nested handler class, the nested handler class being a subclass of a generic handler class and inherits base functionality from the generic handler class, wherein the nested handler class has public access to an enclosing element's private members stored in the private member field of the model element class;

b) providing a singleton model element field handler object comprising a singleton pattern as an abstract base class for handling inlined field values of the model element class configured to access the attribute value stored in the model element class, wherein a first subclass of the model element field handler object comprises a typed model element field handler subclass defining a set value function for setting the attribute value;

c) issuing the set value function to set the attribute value for the model element class;

~~e)d)~~ setting the attribute value;

~~d)e)~~ storing the attribute value in a private member field of the model element class in a same memory block as a declaring class;

f) providing a base filed handler class which acts as an intermediary generic mechanism when setting the filed values in the model element filed handler object, wherein the set value function is configured to:

- i) provide validation of a new value;
- ii) record necessary undo information; and
- iii) dispatch to a sub-classed set field value function in the first subclass which is type safe;

wherein the nested handler class is configured to directly access data in the model element class as the nested handler class has public access to the private members of the model element class; and

wherein the storage of the attribute value is separate from handling of the attribute value.

19. (Canceled)

20. (Previously Presented) The computer-readable medium of claim 1, wherein the singleton pattern enables the data structure to instantiate only one instance of a particular object which is used for supplying functionality for other users who wish to call that one instance.

21. (Previously Presented) The computer-readable medium of claim 9, wherein the singleton pattern enables the data structure to instantiate only one instance of a particular object which is used for supplying functionality for other users who wish to call that one instance.